# FSA Open Source Software Usage Guide

FSA allows controlled usage of open source software. Open source software has many advantages. These include:

- Low initial cost
- Acceptable reliability
- Network friendly
- Simplified license management
- Customizable
- Vendor independence.

FSA should take advantage of the benefits of open source software. While there are benefits to open source software, there are also some disadvantages to be addressed in order to ensure that the usage of open source software does not compromise the existing security architecture. Like the consideration of any new piece of hardware or software, usage of open source software should ultimately depend on the determination of the risks associated with the use of the software.

**Advantages:**

- Low Initial Cost
  The price of an open source program is usually far less than a comparable proprietary program. Although most open source software are freeware, there are some in the market that are not.. Users can either download the software directly from the manufacturer or pay a negligible fee to have a CD-ROM shipped to them. Current users set up distribution networks using community Web sites and CD burners. Their motto is "share and share alike." Open source means anyone can try any program first for free. A user may eventually buy a formal copy (perhaps to get better service), but doesn't have to do so. The software will never expire nor will the programmers responsible for it demand payment.

- Reliability/Stability/Security:
  Open source software may be more reliable and secure than proprietary software. It may not produce as many errors or crash as often as many others (e.g. Linux is known for consistency as it hardly ever crashes.) Due to the fact that any programmer can find and fix bugs in software applications, repairs and improvements can be made rather quickly. The initial program may not be more reliable than a proprietary alternative, but it may mature faster as hundreds or thousands of programmers work daily to correct errors and add new features. This is sometimes referred to as permanent beta testing. The open source community can troubleshoot endlessly to improve software as needed.

  This advantage however depends on the participation of enough competent programmers. Just like proprietary software, the reliability of an open source program depends on clear feedback after rigorous use in a variety of environments. Without

enduring sufficient, talented interest, many open source project have failed. In contrast, proprietary software companies may continue to create and support necessary programs that are not among the favorites for everyone. Some companies are starting to blend the best of both models, by employing a core group of programmers while attracting volunteers from the open source community to test and implement the applications as needed.

- Open source is more network friendly
  Much of the popularity of open source software comes from its performance on the Internet and lesser networks. Open source software can often be networked. For example, more than half the World Wide Web runs on Apache, which is an open source solution. Apple builds its OS X application on BSD, an open source operating system. The Internet is a critical reason why Apple recognizes that they cannot privately innovate Internet functionality as well or as fast as the open source community. Most open source networking solutions are compatible with proprietary software. For example, a lab of Linux computers can be seamlessly nested inside a larger proprietary network using an open source intermediary (e.g. Samba).

- Open source makes license management easier
  License management is much easier with open source. Users are allowed to install unlimited copies, therefore, open source companies do not bother with embedding complicated licenses into their applications (most of the possible licenses favor nonprofit organizations such as schools). There are no risks involved for obtaining illegal copies of the applications nor are their license audits. In addition, no anti-piracy measures are in place (e.g. CD keys, product activation) to further validate software usage.

- Open source can be more customized
  By it's very nature, open source software allows any user or organization with enough expertise to tailor software to their needs. The diversity of Linux distributions reflects this flexibility. Each distribution offers a customized operating system targeting a specific market. There are even two competing graphical desktop interfaces: KDE and GNOME. On the backend, the open, modular nature of open source solutions allows advanced users extraordinary power to customize any aspect of a network (e.g. firewalls, spam filtering, email filtering).

  Beyond customization, anyone can make significant changes to open source code. Proprietary software may offer new features or flexibility in each new version. Although open source software will continue to offer users more rights to the application so as to be able to customize through the so-called right to fork; anyone can take the source code and develop it in a new direction.

- Open source means greater independence from companies/Does not commit FSA to just one vendor
  Open source users have more independence from software companies. Even if a software company goes bankrupt, the community still has the source code. This

independence also means "end of life" decisions or undesirable new features cannot be forced on the users. Systems would not be not locked into a vendor nor are they subjected to using the technical support available with the application. The original programmers may offer the best support but since the software is open anyone can try to improve or support it. FSA can choose the best solution now with the freedom to change in the future. They are not expected to rely on a single vendor for all aspects of support. A comprehensive solution may be easier, but FSA can essentially purchase every component of a solution (including support) from different vendors.

FSA systems that choose to use open source software must peform a risk assessment that includes the below procedures and safeguards.

**Risk Assessment**

An open source software risk assessment must include the following:

- Software Origin
  The system must know the origin of the software. Since it is open source software, there are a variety of programmers working on it and that does leave the potenial for bugs, backdoors or viruses to be coded into the program. The system must know if the website they are getting their software from is reputable within the industry (eg. www.cnet.com) and that the programmers that do the coding also have a reputable track record of developing solid, working products. Identifying other government agencies who use the software is a valid quality control measure.
- Software Compatibility
  Though many open source projects can read proprietary file formats and save documents in formats that proprietary software can read, compatibility issues can still arise, especially in a networked environment where users collaborate across platforms.
- Hardware Compatibility
  Many hardware vendors only provide proprietary drivers for their products, leaving it to the open source community to write their own Linux drivers for new hardware. The system must determine if they have the resources to create any new drivers, etc. for the existing hardware.
- De-centralized Support/ Potential Maintenance Costs
  Without centralized support, users may be left to research solutions to their technical difficulties. Being that the software is free, warranty options are not available, and developers have no legal obligation to replace non-working products. The system must take into consideration the potential cost, if any, of current and future support for the software.
- Service & Support
  Any software solution requires some service and support. For both open source and proprietary software, experts depend on email lists and community Web sites as well as contracted support. The quality and availability of help is proportional to interest and use, especially in open source. The support costs for niche solutions are usually

high. On the backend, open source is common therefore the community is large and offers abundant assistance. A variety of companies offer help for using open source on the backend. On the frontend, open source is often still a niche. So it may be harder to find help (especially in rural areas) and contracted help may be more costly.

- User Friendliness
  The system must take into account the existing technical capabilities of the current technical team. The new software may be easy to master or it may require training. The system must take into consideration potential training costs, if any, to learn how to use the software.

**Testing**

However, in addition to a risk assessment in accordance with the Configuration Management process, a system must perform testing of the software before it goes into production. The software needs to be tested to ensure compatibility with existing FSA software. The content of the software needs to be tested for any bugs/viruses/etc. It is very important to test for these issues ahead of time, because it will be less expensive to address these issues during development, rather than during the implementation or postproduction stages.